Developing a Strategy for Automated Privacy Testing Suites

Ioannis Agrafiotis, Sadie Creese, Michael Goldsmith

Developing a Strategy for Automated Privacy Testing Suites

Ioannis Agrafiotis, Sadie Creese, and Michael Goldsmith

Department of Computer Science
University of Oxford, Oxford, England
{ioannis.agrafiotis,sadie.creese,michael.goldsmith}@cs.ox.ac.uk

Abstract. This paper describes a strategy to develop automated privacy testing suites to assess the correctness of consent and revocation (C&R) controls offered to users by an EnCoRe system[1]. This strategy is based on a formal language in order to provide rigorous and unambiguous consent and revocation specifications, and comprises of two novel procedures that facilitate the process of eliciting testing requirements for privacy properties and creating automated privacy-testing suites. We demonstrate the effectiveness of the strategy by describing our application of the method to a realistic case study, although space limitations preclude a complete presentation.

1 Introduction

The ubiquity of information systems in everyday life has increased the need of individuals to disclose personal information via the Internet, in order to acquire the benefits of today's society. Constantly evolving technologies have on one hand increased the efficiency of the services offered by enterprises, organisations and government institutions, but on the other hand have facilitated them to collect, store, process and share a huge amount of personal data. Concerns about individual privacy are growing mainly because the individuals have little or no knowledge and practical control over how their data is handled by "data controllers". (In this paper we use the term "data controllers" to describe all the parties that handle and process personal data and "data subjects" to describe the individual whose personal data is handled.) The increasing number of incidents where personal data has been lost, used for different purposes, or shared without authority [1], render the use of privacy-enhancing technologies essential for every Internet user.

Although there is no inherent definition to the term privacy [9], the right to privacy has been established in many democratic societies. The difficulties in defining privacy, arise from its complex, multidimensional and context-depended notion. Privacy means different things for different people, and diverse meanings and interpretations derive from peoples' experiences and culture. The volatile notion of

[1] The EnCoRe project [5] is an interdisciplinary research project, a collaboration between UK industry and academia, partially funded by the UK Technology Strategy Board (TP/12/NS/P0501A), the UK Engineering and Physical Sciences Research Council and the UK Economic and Social Research Council (EP/G002541/1).

privacy, the adaptation of new Internet applications such as Web 2.0 and the tendency of the data controllers to use the benefits of the cloud computing, makes the need carefully to study, develop and enforce effective privacy controls for data subjects more urgent than ever.

This paper is inspired by the work undertaken for the needs of the EnCoRe project [5]. We adopt Westin's [8] view of privacy as "the claim of individuals, groups or institutions to determine for themselves when, how and to what extend information about them is communicated to others". Based on this view, in the EnCoRe project we are working to offer the data subject C&R controls through which an individual could manage the flow of their personal data. The overall vision of the project is to "make giving consent as reliable and easy as turning on a tap and revoking that consent as reliable and easy as turning it off again" [5]. To achieve our aim we are addressing challenges at a social, legal, regulatory and technical level.

In this paper we describe a novel strategy for creating test suites to gather evidence of the correctness of the consent and revocation (C&R) controls offered by an EnCoRe system; EnCoRe delivers a range of technology and procedural controls designed to provide data subjects with C&R lifecycle management over their personal data. The strategy combines two procedures: Procedure 1 uses a novel formal language to elicit and document unambiguous requirements, see [2] for discussion of the application of this language; Procedure 2 uses such formal descriptions of requirements to generate a test suite. We have used the novel formal language to formalise requirements for the EnCoRe system in two different case studies. Translating the method into machine-readable language will allow the creation of automated test suites for the EnCoRe system. We demonstrate here our novel strategy by application to an aspect of one of the EnCoRe case studies.

Our intention is to create a testing strategy that could be applied to any system required to handle the life-cycle of consent and revocation controls imposed on data. In this paper the tests are generated for the EnCoRe system.

2 Testing in the EnCoRe System

Our aim is to perform automated tests to ensure correctness of the EnCoRe implementation, by reference to a set of requirements derived from the project scenarios. The strategy will generate tests to assess functional requirements, specifically by focusing on ensuring that the C&R related functions behave as expected. Ultimately enabling us to gain confidence in the integrity of the EnCoRe system. Proving correctness for the EnCoRe system is "elusive"[6] and in this strategy we focus only on privacy requirements. We do not address non-functional requirements here. [2]

The challenges raised in creating privacy test suites for the EnCoRe system are two-fold. They derive from the important role that the data subject has in control-

[2] The neglected non-functional requirements have two different sources. They derive from the assessment of the security properties, which is not relevant to privacy properties and from the complexity of privacy features, such as aggregation or anonymity, which created ambiguities in our attempt to formalise the requirements for the first EnCoRe case study [2].

ling how their personal data is handled by the system, and from the privacy issues that need to be addressed to ensure that the data will be handled in accordance with the data subject wishes expressed in the form of C&R controls.

We use a formal language to describe test requirements rather than natural language in order to provide clear and unambiguous results. This clarity is guaranteed by the existence of a mathematical semantics.

In the literature there are limited references to testing privacy properties. To our knowledge the most comprehensive privacy-testing methodology is that proposed by the Prime Project [4], which champions the development of common criteria and privacy-protection profiles. They propose core privacy properties, well-defined within the academic community, such as: anonymity, unlinkability, unobservability, undetectability, and pseudo-anonymity. However, the assessment of these attributes is still unsuccessful. To our knowledge there are no test suites designed to assess the effectiveness of consent and revocation controls for the handling of personal data, thus we consider our test strategy to be novel.

3 Testing Strategy

The strategy comprises of two novel procedures. The first one aims in eliciting testing requirements based on a formal language, while the second processes the results of the first procedure to generate a list of tests in a machine-readable format suitable for automation within a test harness. We believe that functional testing goes hand-in-hand with the requirements formalisation [2]. The requirements have been identified and expressed using a Hoare logic, namely the C&R logic described in that paper. The C&R Hoare logic enables us to express all the states of a system capable of handling consent and revocation controls. Actions are given in the form of triples that describe a transition from one state to another.

The testing strategy model comprises of initial states, transitions and final states. The states are identifiable, finite in number and expressed with the Hoare Logic.

According to a testing report of the British Computer Society (BCS) [7], a strategy for testing a state transition system should specify:

- The starting state
- The input to that state
- The expected output
- The expected final state

With the C&R Hoare logic we are able to describe with clarity all the aforementioned attributes for a successful testing strategy. The desirable initial state is captured by the pre-condition of the triple, the input that triggers the transition is defined by the action and the expected final state is described in the post-condition. Outputs from the final state, are captured with the form of obligations.

In order to clarify how the requirements are expressed, we explain the notation used. Each action corresponds to a requirement of the following form:

$$\{\textit{pre-condition(rights/permissions)}\}$$
$$\textbf{action}(a, b, \delta)$$
$$\{\textit{post-condition(rights/permissions/obligations)}\}$$

The pre-condition comprises of rights and permissions. Every right consists of a sequence of three letters. The first letter denotes the actor that pertains the specific right, the second letter describes the nature of the right (right to process data or right to share data) and the third letter denotes the data that the right applies to. The permissions are expressed in variables that constrain specific rights.

The action describes a transition from one state of the system to another and denotes the actors that participate in this transition. The first actor is the initiator of the action and the second is the actor influenced by this transition.

The post-condition, in analogy to the pre-condition, comprises of rights and permissions. In addition, it could contain obligations, which are two-folded. They are actions that need to be triggered in the future, under certain conditions or actions that should be cascaded to third parties in order for the post-condition to be completed. In the latter case, a third actor is also influenced by the transition from one state to another.

The state of the system comprises of:

- Actors,
- Rights, predicates of the logic that are either true or false
- A number of Consent and Revocation variables, that define the dimensions in which restrictions can be imposed on data use
- The actual values of these C&R variables

Each action can be triggered when the pre-condition is met and when completed could either

1. alter rights on one actor
   - alter rights on more than one if there exists an obligation in the post-condition
2. update data
3. change variables
4. set notification rules
5. send notifications

3.1 Procedure 1: Eliciting test requirements

With the procedure illustrated below, we elicit the requirements for the testing suites. Based on the formalisation of the system's requirements, we analyse every requirement and derive the factors and the results that will define the test suites on the next procedure. More specifically, we identify the actors of the system and those that participate in the specific formalisation, we clarify the rights that are

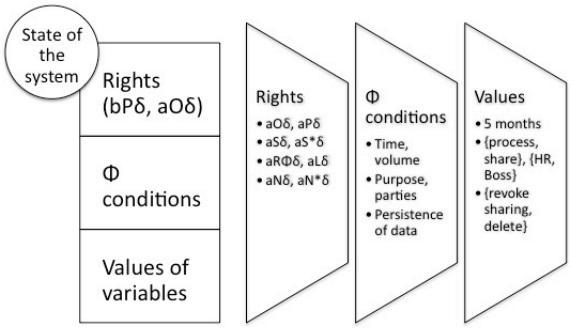

Fig. 1. The state of the system

altered by the action and the values of the variants that restrict every right. Finally, any notifications or obligations that require further action from third parties, are distinguished and taken under consideration.

Below we illustrate the schema of the first model:

1. Identify the actors of the system (from the state of the system)
2. Identify actors participating in the action ($\mathbf{grant}(\mathbf{a}, \mathbf{b}, \delta) \implies$ that the actors involved are a and b)
3. Identify whose actor's rights are influenced by the action ($\mathbf{grant}(\mathbf{a}, \mathbf{b}, \delta) \implies$ that the actor whose rights are influenced is b)
   (a) Check for obligations in the post condition (for example in the cascading revocation action rights are reduced from both data controller and third parties)
4. Identify the class that the action belongs to. We have identified five different classes, namely:
   (a) **grant** actions
   (b) **revoke, delete** actions
   (c) **notify** actions
   (d) **change consent and revocation variables** actions
   (e) **update** actions
5. Identify which rights are influenced from the action. According to the class that actions belong, rights may be added, reduced or remain the same.
6. Identify the variables of consent and revocation that are influenced and the attributed values that these have. According to the class that actions belong to, variables could be added, subtracted or change their values.

3.2 Procedure 2: Producing test suites

The factors and results identified from the previous procedure, are used as variables that influence the creation of the test suites. In order to produce test suites we need to consider the following:

1. Verify that the pre-condition of the action is true
2. Verify that **only** the identified actor has been influenced from the action and **not** the other actors of the system
3. The rights have been altered appropriately when actions belong to the a, b and d class, data changed when actions belong to the c class and variables changed when actions belong to the e class
4. No more/less rights have been added/reduced to/from the actors
5. The values of the variables that constrain the rights are respected by actors
6. Notification was sent/ not sent to the appropriate actor after the action was completed

The test suites are designed to exercise "valid transitions" between states. Since we have formalised the final state of the system, based on Hoare rules and axioms we could test only the actions that are allowed to be triggered from that state. However, test cases may also be designed to test that "unspecified transitions" cannot be triggered [7]. This distinction allows our testing strategy to be simple by only testing the valid transitions or more thorough by verifying that transitions prohibited by the Hoare rules, are also denied by the implementation of the system.

The test suites generated for each action of the Hoare logic provide a black box of tests, meaning that every time a specific action is triggered and the transition from one state to another is completed, the tests required to validate the correctness of such a transition will remain the same.

There are though limitations to our model. For testing sequential actions 1,2,3 leading from state A of the system to state D, the system should generate tests for each action separately, testing the transition from state A to B, from B to C and from C to D.

Concurrency of actions is another limitation. When two or more actions are triggered simultaneously, or an action is triggered before the system has reacted on a previous transition, the testing suites generated by our model should be complemented with further tests to provide efficient assessment.

4 Applying the Testing Strategy to a Real Case Study

The case study selected for the validation of the models and the logic is the Enhanced Employee Data Scenario [2]. Our choice was informed by the fact that the management of employee data in organisations is a well-understood problem, and employees' privacy offers interesting issues in terms of managing consent and revocation controls in a context where different business, legal and personal requirements need to be taken into account. The case study describes a number of use case scenarios and elicits from these a list of requirements. We explore how we can automate

the process for creating test suites to ensure the correctness of the invoking consent and revocation controls.

We have generated tests for all the different use cases of the scenario but for the purpose of this paper we will apply our strategy and develop the testing requirements for two use cases only. In the first use case, Mary (our data subject) has just been hired by a company X. In the second use case, Mary resigns from company X. These use cases provide complex situations and conflicting actions, since in the first use case Mary is consenting to the use and share of her data, whereas in the second Mary requests her data to be deleted.

Testing verifies that the interactions of the system under examination, in this particular case EnCoRe system, with the environment through "points of control and observation" [6] conform with specifications. The specifications of the EnCoRe system, for this particular case study, have been identified and formalised in [2]. We need to capture the environment of the system and according to Armando et al [3], in order to describe the testing environment we need to define the System Under Test (SUT). The actors that use the specific EnCoRe system are defined as SUT and all the other actors are simulated by the tester. In our case study, the SUT is HR department of the X company, which is represented by the actor h in the system model. The actors Mary, Mary's boss and third parties are all simulated by the tester and comprise the environment of the system.

4.1 Mary is hired by company X

Before she starts in her new position she reports to Human Resources (HR) where she fills out various forms, including necessary health information. She signs a form agreeing to the terms and conditions which are stored by HR.

$$\{mO\delta\}$$
$$\mathbf{grant}^{\dagger}(m, h, \delta, \Phi)$$
$$\{hL\delta \wedge hP\delta \wedge hS^{*}\delta\}$$

where Φ = destination:$\Pi \wedge \neg\pi \wedge$ purpose:$p \wedge$ time duration:$t \wedge$ times processed:t^{*} and $\Pi \subseteq$ {Mary's boss} , $\pi \subseteq$ {third parties}and $p \subseteq$ {internal purposes}, $t \subseteq$ {One year - five years} and $t^{*} \subseteq$ {One time - 100 times}

The semantics of the formalisation are: m= Mary, h= HR department, δ= health information. Π is a variable that allows the HR department to share data only with Mary's boss, π restrains the HR department from sharing Mary's data with third parties while p defines the purpose for which the data should be processed. Furthermore, there are variables describing the duration of consent, t denotes the years that data should be stored for, and t^{*} the number of times the HR department may process the data.

All the actions performed in the system are defined by the system administrator and in this specific case study, the HR department. Thus, the options from which Mary can choose from are pre-defined by the HR department. Furthermore, all the

actions may invoke changes in the rights of the actors. In the above formalisation, before the action Mary was the owner of the data. After the action, there is a transition to a state where the HR department possess the right to process Mary's data ($hP\delta$), the right to store Mary's data ($hL\delta$) and the right to share Mary's data($hS\delta$), all of which are restrained by conditions described in Φ.

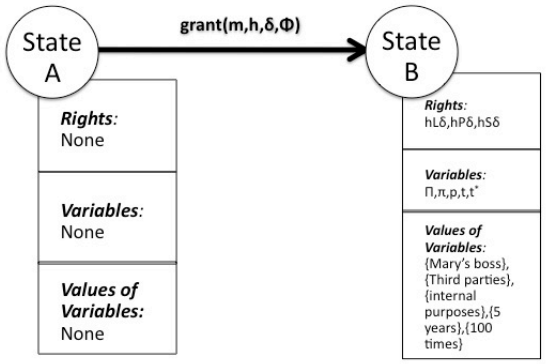

Fig. 2. Transition from the initial state to the final by the grant action

By applying the first procedure, we identify the actors of the system, the rights that each actor has, the variables that restrict the rights and which rights should be altered. In essence, we elicit requirements that allows us to define the initial state, the triggering action and the final state of the system.

The actors of the system are four, namely:

1. Mary
2. HR department
3. Mary's boss
4. Third party

From these actors, those implicated into the action are Mary and the HR department. The latter actor is influenced by this action. The action is part of the "grant" class and there are no obligations created. With this action the rights that are influenced are three, namely:

1. Right to collect
2. Right to process

3. Right to share

The data pertaining to the action is Mary's health data and the variables defined are five, namely:

1. Destination (Mary's boss) and NOT (Third parties)
2. Purpose (internal use)
3. Time of consent (Up to five years)
4. Times data processed (up to 100 times)

These results are presented in the table below:

Table 1. Results from the application of the first procedure

Actor	**Rights on data**	**Rights to change by the action**
Mary	owner of the data	none
HR department	none	store, process, share
Mary's boss	none	none
Third parties	none	none

We then apply Procedure 2 to the results of Procedure 1, recorded in Table 1, and produce the test requirements described in Table 2 below.

We need to verify that only the HR department has obtained the appropriate rights and that Mary's choices are enforced. Thus, we create tests to verify whether the HR department has obtained the right to collect, process and share Mary's data and if her choices (variables) are enforced. Further testing is required to examine if the HR department has obtained more rights than those mentioned and if any other actor of the system was influenced. The last tests aim at verifying that transitions prohibited in the Hoare logic are not allowed by the implementation.

Our aim is that the tests, captured in this procedure, will be produced automaticaly. There is an ongoing work to develope an algorith that, based on the logic, will create tests whenever an action occurs in the system. Furthermore, the algorith will be configured and will produce different levels of testing according to how thorough the company would like the testing procedure to be.

Table 2. Test cases generated by the application of the second model

Actor	Test	Pre-condition	Post condition
Third party	Attempt to access Mary's data as other users	No Access	No Access
Mary's boss	Attempt to access Mary's data held within the HR department	No Access	No Access
HR department	Attempt to process Mary's data for 99th time	No Access	Access Granted
HR department	Attempt to process Mary's data for 101th time	No Access	No Access
HR department	Attempt to process Mary's data after five years	No Access	No Access
HR department	Attempt to share Mary's data with Mary's boss	No Access	Access Granted
HR department	Attempt to notify Mary	No Access	No access
HR department	Attempt to release Mary's data to third parties	No Access	No Access

Each row effectively describes a single test of the system: a test harness must first establish a correct system state (satisfying the pre-condition), attempt the access specified, and then observe whether the resulting system state meets the postcondition. If it does, the test is considered to be passed; if not, a failure is registered. Of course, given all the unconstrained factors that contribute to the system state, there is no guarantee that this result is completely determined; the same test might give the opposite result in other circumstances. The role of testing within EnCoRe, however, is to contribute to evidence-gathering about the correctness of the implementation, not to be the sole arbiter, and tests passed will contribute to confidence.

The testing suites are executed twice in every transition. Firstly, we test if the triggering of the action from the initial state was valid. The expected result for the tests is described in the third column of the Table 2 above. The second set of tests aim to verify that the transition has resulted in reaching the desirable final state and the result for each test is defined in the fourth column of the Table 2.

4.2 Mary Leaves the Company

Mary decides to leave the company. She wishes to revoke her consent regarding the use of her data and requires all data to be deleted.

$$\{mO\delta \wedge hL\delta \wedge hP\delta \wedge hS\delta \wedge hR\delta\Phi\}$$
$$\mathbf{delete}(m, h, \delta)$$
$$\{\neg hL\delta \wedge \neg hP\delta \wedge \neg hS\delta\}$$

When the transition is completed, in the final state of the system the only actor with rights should be Mary.

The semantics of the formalisation are: m= Mary, h= HR department, δ= health information. $hR\delta\Phi$ is a right denoting that the HR department respect the choices ($\Phi variables$) that Mary gave in the past and restrict the process and sharing of her data.

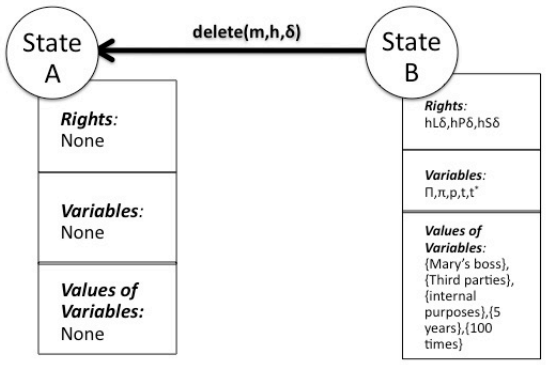

Fig. 3. Transition from the initial state to the final by the delete action

By applying the first procedure, we elicit the requirements presented below:
The actors of the system are four, namely:

1. Mary
2. HR department
3. Mary's boss
4. Third party

From these actors, those implicated into the action are Mary and the HR department.The latter actor is influenced by this action. The action is part of the "revoke" class, meaning that the rights are subtracted and there are no obligations created. With this action the rights that are subtracted from the HR department are three, namely:

1. Right to collect
2. Right to process
3. Right to share

The data pertaining to the action is Mary's health data and the variables that were defined by Mary in previous transitions are five, namely:

1. Destination (Mary's boss) and NOT (Third parties)
2. Purpose (internal use)
3. Time of consent (Up to five years)
4. Times data processed (up to 100 times)

These results are presented in the table below:

Table 3. Results from the application of the first procedure

Actor	**Rights on data**	**Rights to change by the action**
Mary	owner of the data	owner of the data
HR department	store, process, share	none
Mary's boss	none	none
Third parties	none	none

We then apply Procedure 2 to the results of Procedure 1, recorded in Table 3, and produce the test requirements described in Table 4 below.

We need to verify that the appropriate rights were revoked by the HR department and that Mary's data is deleted. Thus, we create tests to verify whether the HR department still possesses the right to collect, process and share Mary's data. Further testing is required to examine if the HR department has any other rights and if any other actor of the system was influenced.

Table 4. Test cases generated by the application of the second model

Actor	**Test**	**Pre-condition**	**Post condition**
Third party	Attempt to access Mary's data as other users	No Access	No Access
Mary's boss	Attempt to access Mary's data held within the HR department	No Access	No Access
HR department	Attempt to process Mary's data for 99th time	Access Granted	No Access
HR department	Attempt to process Mary's data for 101th time	No Access	No Access
HR department	Attempt to process Mary's data after five years	No Access	No Access
HR department	Attempt to share Mary's data with Mary's boss	Access Granted	No Access
HR department	Attempt to notify Mary	No Access	No access
HR department	Attempt to release Mary's data to third parties	No Access	No Access

The test suites designed for this formalisation and presented in Table 4 are the same with the previous use case, but the results of the tests are different. The testing is successful because the actions described transitions from state A to state B and backwards. Since the transitions were only between these two states of the system, the tests should remain the same, since the pre-condition for the one transition was the post-condition of the other and vice versa.

5 Conclusion and future work

With the development of Internet applications such as Web 2.0 or cloud computing, new challenges arise for protecting individuals' privacy. Research is conducted in diverse disciplines examining the legal, regulative and technical aspects of privacy, but little has been achieved so far regarding the development of a methodology for testing privacy properties.

In this paper, we proposed a strategy comprised by two novel models, that develops automated testing suites for effectively validating the correctness of consent and revocation controls. We applied the strategy on a real case scenario to prove its practicality, at least as far as individual actions are concerned.

For future work, we intend to consider the effects of concurrency and how our test strategy should evolve to probe them. We will also validate our method on another case study in order to create a detailed assessment of the applicability of the approach.

References

1. I. Agrafiotis, S. Creese, M. Goldsmith, and N. Papanikolaou. Reaching for informed revocation: Shutting off the tap on personal data. *Privacy and Identity Management for Life*, pages 246–258, 2010.
2. I. Agrafiotis, S. Creese, M. Goldsmith, and N. Papanikolaou. Applying formal methods to detect and resolve ambiguities in privacy requirements. *Privacy and Identity Management for Life*, pages 271–282, 2011.
3. A. Armando, R. Carbone, L. Compagna, K. Li, and G. Pellegrino. Model-checking driven security testing of web-based applications. In *Software Testing, Verification, and Validation Workshops (ICSTW), 2010 Third International Conference on*, pages 361–370. IEEE, 2010.
4. Marco Casassa Mont, Siani Pearson, Gina Kounga, Yun Shen, and Pete Bramhall. Privacy and identity management in europe.overview of existing assurance methods in the area of privacy and it security. Technical report, HP Labs, Bristol, 2004.
5. http://www.encore-project.info.
6. J.C. Fernandez, C. Jard, T. Jéron, and C. Viho. An experiment in automatic generation of test suites for protocols with verification technology* 1. *Science of Computer Programming*, 29(1-2):123–146, 1997.
7. British Computer Society Specialist Interest Group in Software Testing (BCS SIGIST). Standard for software component testing. Technical report, British Computer Society, Working Draft 3.4, 2001.
8. A.F. Westin. *Privacy and freedom*, volume 97. London, 1967.

9. Edgar A. Whitley. Information privacy consent and the 'control' of personal data. *Inform. Secur. Tech. Rep.*, DOI:10.1016/j.istr.2009.10.001, 2009.